10. Les tableaux statiques / dynamiques

Définition : Tableau

Au contraire d'une variable ordinaire, un **tableau** est une variable qui peut contenir **plusieurs valeurs**, à condition que ces valeurs **soient du même type**.

En CSharp il existe 3 types de tableaux :

- Les tableaux statiques de dimension 1
- Les tableaux statiques de plusieurs dimensions (bien qu'on dépasse très rarement la dimension 2)
- Les tableaux dynamiques

L'utilisation des tableaux passe par 3 étapes :

1- Déclaration
2- Allocation de l'espace mémoire nécessaire pour le stockage
3- Lecture / Ecriture des valeurs

1. Les tableaux statiques de dimension 1

Définition

Un tableau de dimension 1, qui contiendrait **n** valeurs, peut être vu de la manière suivante :

v_1	v_2	…	v_n

Syntaxe d'utilisation d'un tableau de dimension 1 :

1- Déclaration

```
type[] nom_tableau;
```

Exemple :

```
byte[] t1;
```

2- Allocation de l'espace mémoire nécessaire pour le stockage

```
nom_tableau = new type[taille];
```

Exemple :

```
t1 = new byte[4];
```

3- Lecture / Ecriture des valeurs

Pour lire ou écrire dans un tableau il faut le parcourir case par case.
Les cases du tableau sont indexées en commençant par **0**.
Les indices des cases d'un tableau de **N valeurs** sont entre **0** et **N-1**.

Indices	**0**	**1**	…	**n-1**
Valeurs	v_1	v_2	…	v_n

```
nom_tableau[indice] = valeur;
```

Exemple :

```
t1[0] = 10;
t1[3] = 17;
t1[1] = t1[3] + t1[0];
t1[2] = 255;
```



REMARQUES

- Il est possible de **rassembler ces 3 étapes** en **une seule instruction** dans la syntaxe suivante :

  Syntaxe 1 : `type[] nom = new type[]{V1, V2, …, Vn};`

  Ou bien

  Syntaxe 2 : `type[] nom = {V1, V2, …, Vn};`

  Exemple :

  ```
  int[] T = new int[]{10000, 12, -5, 9, 8853};
  ```

  Ou bien

  ```
  int[] T = {10000, 12, -5, 9, 8853};
  ```

- Toute séparation dans la syntaxe **2** provoque **une erreur de compilation**.

  Exemple :

  ```
  int[] T;
  T = {10000, 12, -5, 9, 8853};
  ```

- Dans la syntaxe **1**, on peut ajouter la taille du tableau, mais si la taille est différente du nombre de valeurs spécifiées, **une erreur de syntaxe** est provoquée.

  Exemple :

  ```
  int[] T = new int[5]{10000, 12, -5, 9, 8853};
  ```

- L'attribut ***length*** donne la taille (le nombre de cases) du tableau
  Syntaxe : **nom_tableau.length**
- L'indice d'une case **doit être compris** entre **0** et **taille - 1**. Autrement, une exception de type ***IndexOutOfRange*** est provoquée.

```
int[] t= { 1,2 };
Console.WriteLine(t[2]);
```

L'exception IndexOutOfRangeException n'a pas été gérée

Une exception non gérée du type 'System.IndexOutOfRangeException' s'est produite dans CoursCSProjet.exe

Informations supplémentaires : L'index se trouve en dehors des limites du tableau.

- Si une case n'est pas remplie, alors elle contiendra la valeur par défaut correspondante au type du tableau :
  Chiffres : **0**
  Caractères : **le caractère vide**
  Chaines de caractères : **la chaine vide**

Exemple

```
string[] T = new string[2];
byte i = 0;
// remplissage
Console.WriteLine("tapez le nom d'une ville");
T[i] = Console.ReadLine();
i++;
Console.WriteLine("tapez le nom d'une ville");
T[i] = Console.ReadLine();

// affichage
i = 0;
Console.WriteLine("\nles villes que vous avez tapees :");
Console.Write(T[i] + "  ");
i++;
Console.Write(T[i] + "  ");
```

file:///C:/Users/s_elyahyaoui/Documents/Visual Studio 2015/Projects/CoursCSPr...

```
tapez le nom d'une ville
casablanca
tapez le nom d'une ville
rabat

les villes que vous avez tapees :
casablanca  rabat
```

Exercice 1

Quel est le résultat d'exécution de ce programme :

```
char t1[] = {'b', 'o', 'n', 'j', 'o', 'u', 'r'};
char t2[] = {'v', 'i', 'n', 'c', 'i'};
for (int i=0 ; i<t1.Length ; i++) Console.Write(t1[i]);
for (int i=0 ; i<t2.Length ; i++) Console.Write(t2[i]);
```

Exercice 2

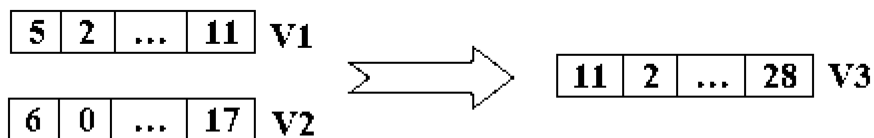
Écrire un programme qui demande à l'utilisateur de saisir les notes d'examen de **10** élèves, qui seront stockées dans le tableau appelé **notes**. Le programme doit ensuite afficher le nombre d'élèves qui ont la moyenne.

Exercice 3

En utilisant une boucle **do-while**, remplir un tableau d'entiers par les **10 premières puissances de 2** : (**2**, **4**, **8**, **16**, **32**, …, **1024**).
Afficher ensuite les éléments du tableau séparés par des tabulations :
2 4 8 16 32 64 128 256 512 1024

Exercice 4

Ecrire un programme qui lit les éléments des tableaux d'entiers **V1** et **V2** de même taille **N** qui sera lue au clavier, et qui stocke **les sommes** des éléments de **V1** et **V2** dans un autre tableau **V3** :



Afficher ensuite les éléments de **V3**

2. Les tableaux statiques de dimension 2

Définition

Un tableau de dimension 2, peut être vu comme une grille, contenant des lignes et des colonnes.
Un tableau de **4 lignes et 6 colonnes** peut être vu de la manière suivante :

ligne 0						
ligne 1						
ligne 2						
ligne 3						
colonnes:	0	1	2	3	4	5

Syntaxe d'utilisation d'un tableau de dimension 2 :

1- Déclaration

```
type[,] nom_tableau;
```

Exemple : `byte[,] t1;`

2- Allocation de l'espace mémoire nécessaire pour le stockage

```
nom_tableau = new type[nbr_lignes,nbr_colonnes];
```

Exemple : `t1 = new byte[3,4];`

3- Lecture / Ecriture des valeurs

Pour lire ou écrire dans un tableau il faut le parcourir case par case.
Les cases du tableau sont désignées par l'indice de ligne et l'indice de colonne.
Les lignes et colonnes sont indexées en commençant par **0**.

```
nom_tableau[ligne,colonne] = valeur;
```

Exemple :

```
// ligne 1
t1[0,0] = …;
t1[0,1] = …;
t1[0,2] = …;
t1[0,3] = …;
// ligne 2
t1[1,0] = …;
t1[1,1] = …;
t1[1,2] = …;
t1[1,3] = …;
// ligne 3
t1[2,0] = …;
t1[2,1] = …;
t1[2,2] = …;
t1[2,3] = …;
```



REMARQUES

- Il est possible de **rassembler ces 3 étapes** en **une seule instruction** dans la syntaxe suivante :
  `double[,] reels = new double[,]{ {1.1, 27.2}, {15.07, 81.3} };`
- Les remarques précédentes sont valables aussi pour les tableaux de dimension 2.

Exemple : remplir un tableau d'entiers de 3 lignes et 2 colonnes depuis le clavier

```
int[,] t1 = new int[3,2];
int L = 0, C;
while (L < 3)
{
    Console.WriteLine("Remplir ligne " + (L + 1));
    C = 0;
    while (C < 2)
    {
        Console.Write("tapez une valeur\t");
        t1[L,C] = int.Parse(Console.ReadLine());
        C += 1;
    }
    Console.Write("\n");
    L += 1;
}
```

```
file:///C:/Users/s_elyahyaoui/Documents/Visual Studio 2015/Projects/CoursCSPr...
Remplir ligne 1
tapez une valeur        4
tapez une valeur        2

Remplir ligne 2
tapez une valeur        6
tapez une valeur        18

Remplir ligne 3
tapez une valeur        5
tapez une valeur        100
```

Exercice 1

Remplir deux matrices **M1** et **M2** de **2** lignes et **2** colonnes depuis le clavier, et créer la matrice **S** qui représente leur somme.
Afficher ensuite les éléments de **S**.
Exemple :

M1

1	1
1	1

M2

4	4
4	4

S

5	5
5	5

Exercice 2

Remplir une matrice nommée **A** comme ceci :

1	1	1	1	1	**5 1ières** puissances de **1**
2	4	8	16	32	**5 1ières** puissances de **2** **(2^1, 2^2, 2^3, …)**
3	9	27	81	243	**5 1ières** puissances de **3** **(3^1, 3^2, 3^3, …)**
4	16	64	256	1024	**5 1ières** puissances de **4** **(4^1, 4^2, 4^3, …)**

Afficher ensuite la moyenne de chaque ligne comme ceci :

```
Moyenne de la ligne 1 :   1.0
Moyenne de la ligne 2 :   12.4
Moyenne de la ligne 3 :   72.6
Moyenne de la ligne 4 :   272.8
```